Eliciting and Specifying Quality Attribute Requirements

Rob Wojcik
Senior Technical Staff

Rob is a senior member of the technical staff in the Research, Technology, and System Solutions Program at the Carnegie Mellon University's Software Engineering Institute (SEI), a position he has held since 2004. In his current position, he performs training and consulting in software architecture technology and software architecture evaluations.

© 2013 Carnegie Mellon University

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
16 JAN 2013		**00-00-2013 to 00-00-2013**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Eliciting and Specifying Quality Attribute Requirements	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Carnegie Mellon University,Software Engineering Institute,Pittsburgh,PA,15213**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**27**	

What we'll discuss today

Quality Attributes

Eliciting Quality Attribute Requirements

Quality Attribute Workshop

Quality Attribute Scenarios

I'll take questions at the end of the presentation.

© 2013 Carnegie Mellon University

Polling Question

When is the best time to specify quality attribute (non-functional) requirements?

1. After the software architecture is established
2. Before the software architecture is established
3. Quality attributes are not important

© 2013 Carnegie Mellon University

What are Quality Attributes?

Measurable or testable properties of a system used to indicate how well the system satisfies the needs of its stakeholders

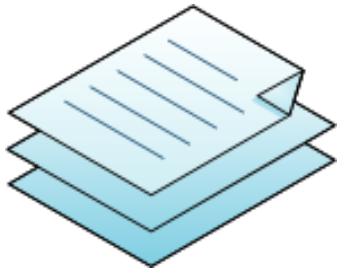
Here are some examples of quality attributes:

- availability
- configurability
- modifiability
- performance
- reliability
- reusability
- security
- throughput

© 2013 Carnegie Mellon University

Stakeholders and Quality Attributes

Lead To



Stakeholder Needs

Quality Attribute Requirements

"Increase market share" ----------> Modifiability, Usability

"Maintain a quality reputation" ----------> Performance, Usability, Availability

"Introduce new capabilities seamlessly" ----------> Performance, Availability, Modifiability

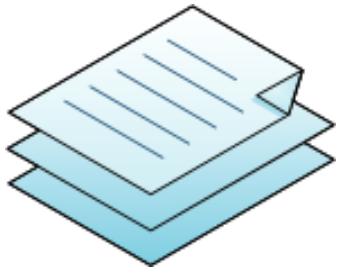
"Provide a programmer-friendly framework" ----------> Modifiability

"Integrate with other systems easily" ----------> Interoperability, Portability, Modifiability

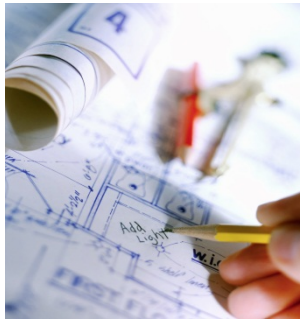
© 2013 Carnegie Mellon University

Quality Attributes and Architecture

The degree to which a system satisfies quality attribute requirements is directly dependent on architectural structure.



Key Input Into



Quality Attribute Requirements

Software Architecture Design

Consequently, architects need to have a solid understanding of the quality attribute requirements for a system when they are designing the system's software architecture.

© 2013 Carnegie Mellon University

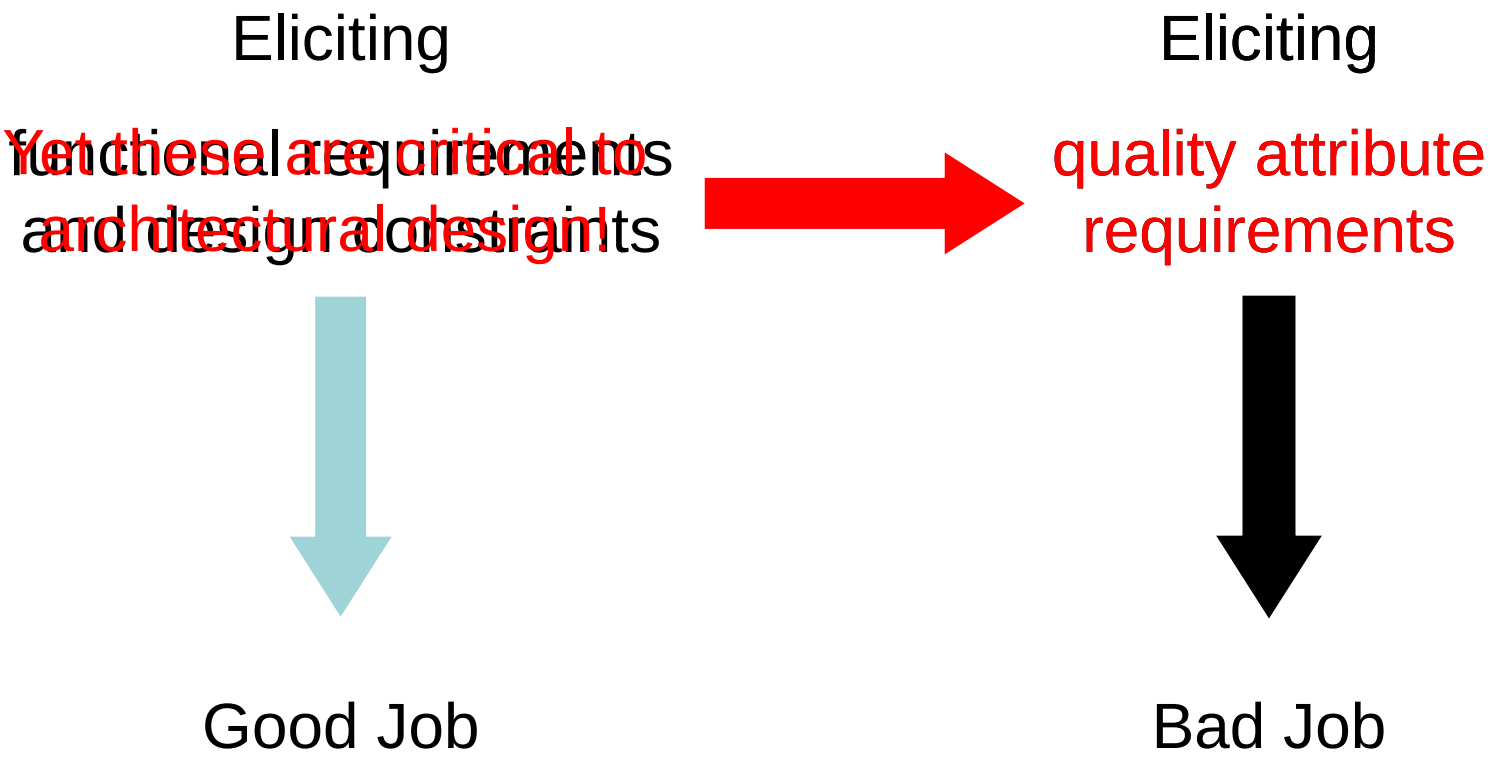
Polling Question

What approach does your organization use to specify quality attribute requirements?

1. We ask management what they think the system should do.
2. We discuss the system's quality attributes once the system is designed.
3. We use a method to gather the views of all our stakeholders early in the development life cycle.
4. We don't worry about quality attributes.

© 2013 Carnegie Mellon University

Eliciting Quality Attribute Requirements



© 2013 Carnegie Mellon University

Problems With Quality Attribute Requirements

Non-Operational requirements

- “The system must be easy to use.”
- “The system must have high performance.”
- “The system must be portable.”

Debating the quality attribute to which a system behavior belongs

- “The system must process 10,000 messages per second.”

Vocabulary variations

- Everyone knows what “high performance” means, right?

© 2013 Carnegie Mellon University

Quality Attribute Workshop (QAW)

Facilitated method

- system-centric
- used before the software architecture has been created

Engages system stakeholders early in the life cycle

Reveals the driving quality attribute requirements of a software-intensive system

- scenario based

A QAW delivers the quality attribute requirements for the system, documented as refined and prioritized quality attribute scenarios

The quality attribute scenarios can then be used as the basis for designing the software architecture for the system.

© 2013 Carnegie Mellon University

QAW Steps

1. QAW Presentation and Introductions
2. Business/Programmatic Presentation
3. Architectural Plan Presentation
4. Identification of Architectural Drivers
5. Scenario Brainstorming
6. Scenario Consolidation
7. Scenario Prioritization
8. Scenario Refinement

© 2013 Carnegie Mellon University

QAW Steps

1. QAW Presentation and Introductions
2. Business/Programmatic Presentation
3. Architectural Plan Presentation
4. Identification of Architectural Drivers
5. Scenario Brainstorming
6. Scenario Consolidation
7. Scenario Prioritization
8. Scenario Refinement

Today's focus

© 2013 Carnegie Mellon University

Step 4: Identification of Architectural Drivers

The QAW facilitators identify the architectural drivers that are key to realizing quality attribute goals by

- presenting a distilled list of the architectural drivers they heard during the Business/Programmatic and Architecture Plan presentations
- asking for clarifications, additions, or deletions to reach a consensus on the architectural drivers

The final list of architectural drivers focuses the stakeholders during scenario brainstorming.

© 2013 Carnegie Mellon University

Step 5: Scenario Brainstorming

Stakeholders generate scenarios using a facilitated brainstorming process.

Each stakeholder either generates a scenario in round-robin fashion or opts to pass.

Each stakeholder may have an opportunity to contribute more than one scenario, depending on the number of stakeholders in the QAW and the allocated time for the workshop.

© 2013 Carnegie Mellon University

Step 5: Scenario Brainstorming

Scenario brainstorming guidance

- Quality attribute requirements are most effectively characterized vis-à-vis scenarios.
- Scenarios are "short stories" that describe a system interaction with respect to some quality attribute.
- Well-formed scenarios have a stimulus, an environment, and a response.
- The QAW focuses on three kinds of scenarios:
  - use case – anticipated uses of the system
  - growth – anticipated changes to the system
  - exploratory – unanticipated stresses to the system (uses and/or changes)

© 2013 Carnegie Mellon University

Stimuli, Environment, Responses

Use case scenario

> *A remote user requests a database report via the Web* **during a peak period** and receives it within 5 seconds.

Growth scenario

> *Add a new data server* **to reduce latency in the use case scenario to 2.5 seconds** within 1 person-week.

Exploratory scenario

> *Half of the servers go down* **during normal operation** without affecting overall system availability.

© 2013 Carnegie Mellon University

Step 6: Scenario Consolidation

The QAW facilitators ask stakeholders to identify those scenarios that are very similar in content.

- Similar scenarios are merged to prevent a “dilution” of votes when voting is done in the next step.
- QAW facilitators attempt to reach a consensus with the stakeholders before merging scenarios.

© 2013 Carnegie Mellon University

Step 6: Scenario Consolidation

Example of scenario consolidation

Scenarios that are similar in content are grouped together.

- In the event of a processor fault, the system can be rebooted/reinitialized.
- A processor failure or crash doesn't adversely affect any other components (no second-order failures).
- Software continues to operate even if the host fails (mission computer).

© 2013 Carnegie Mellon University

Step 7: Scenario Prioritization

Each stakeholder is allocated a number of votes equal to approximately 30% of the number of scenarios generated.

- The actual number of votes allocated to stakeholders is rounded up to an even number of votes.
- Voting occurs in two rounds where each stakeholder allocates exactly half of his or her total votes in each round.
- Stakeholders can allocate any number of votes to any scenario they like.
- Votes are counted, and the scenarios are prioritized accordingly.

© 2013 Carnegie Mellon University

Step 8: Scenario Refinement

The top scenarios are further refined.

- Typically, the top five scenarios are refined, but the exact number will depend on the time available.

The QAW facilitators further elaborate each scenario and

- document the business/programmatic goals affected by the scenario
- describe the relevant quality attributes
- rephrase it in six parts: a stimulus, a stimulus source, an environment, an artifact, a response, and a response measure
- document a list of related questions that stakeholders want to ask
- document any issues that may arise during scenario refinement

© 2013 Carnegie Mellon University

Example Scenario Refinement - 1

Scenario	The track capacity is saturated during peak operations over a large-sized theatre and degrades in a predictable and useful manner.
Business Goals	Mission effectiveness
Quality Attributes	• performance • scalability

© 2013 Carnegie Mellon University

Example Scenario Refinement - 2

Stimulus	Some resource capacity is saturated (or hits high watermark).
Stimulus Source	Network, memory, applications, and so on
Environment	Operational with high load conditions (or battle damage)
Artifact	Track Manager or host system
Response	• throttling appropriately • request for or release of resources
Response Measure	Predictable behavior as appropriate per platform

© 2013 Carnegie Mellon University

Example Scenario Refinement - 3

Questions	What is the track capacity? Is it based on hardware? How do you know it's saturated? In what ways can you degrade the quality of service associated with the tracks? What can be automated? Which architectural decisions apply and support that automation? Which degrade modes do stakeholders want to see implemented?

© 2013 Carnegie Mellon University

Example Scenario Refinement - 4

Issues	We can't size systems *a priori* to eliminate oversaturation. Track Manager needs a way to compensate for saturation in a doctrinally appropriate manner. Open a dialogue with the applications using Track Manager to identify objects of interest, so that applications can have a say in this answer.

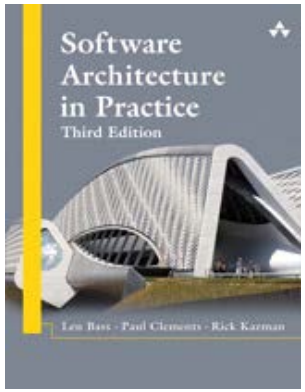
© 2013 Carnegie Mellon University

For More Information

Contact me at

Rob Wojcik
rwojcik@sei.cmu.edu

- Find out more about the QAW: www.sei.cmu.edu/architecture/tools/establish/qaw.cfm
- For more about the SEI approach to quality attributes and architecture-centric engineering, start exploring at www.sei.cmu.edu/architecture/



- Also see *Software Architecture in Practice, 3rd edition* written by Len Bass, Paul Clements, & Rick Kazman and published by Addison-Wesley as part of the SEI Series in Software Engineering.
- Visit www.sei.cmu.edu/library/abstracts/books/9780321815736.cfm

© 2013 Carnegie Mellon University

NO WARRANTY

THIS MATERIAL OF CARNEGIE MELLON UNIVERSITY AND ITS SOFTWARE ENGINEERING INSTITUTE IS FURNISHED ON AN “AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this presentation is not intended in any way to infringe on the rights of the trademark holder.

This Presentation may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

© 2013 Carnegie Mellon University

SATURN 2013

Software Engineering Institute | Carnegie Mellon

Minneapolis, Minnesota

April 29 to May 3, 2013

www.sei.cmu.edu/saturn/2013

© 2013 Carnegie Mellon University